ГУАП

КАФЕДРА № 34

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Жиданов К.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
|  |
| по курсу: |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 3145 |  |  |  | Калинин Е.Д. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

**Задание 1:** написать программу, реализующую заданный алгоритм:

Посчитать количество ненулевых элементов в массиве.

**Ход выполнения:**

1. Чтобы посчитать количество ненулевых элементов в массиве нужно сравнить эти элементы с нулем. Если элемент будет отличаться от нуля, то мы увеличиваем количество на 1
2. Регистры для ввода и вывода данных:

BL – счетчик элементов

DL – пороговое значение

1. Заносим программу в отладчик
2. Выполняем программу, подавая на вход различные тестовые наборы, после каждого запуска убеждаясь, что результат работы программы является корректным и совпадает с полученным при выполнении вручную.

**Код программы:**

xor eax, eax

mov dl,0

lea bp,[array] ;Загружаем адрес массива

xor bl,bl ;Обнуляем счётчик элементов

mov cx,4;Устанавливаем значение счётчика цикла

next:

mov al, [array] ;Загружаем очередной элемент

cmp al,dl ;Сравниваем элемент с пороговым значением

jg skip ;Пропускаем элемент, если значение не подходит

inc bl ;Увеличиваем счётчик элементов

skip:

inc bp ;Перемещаем указатель на следующий элемент

loop next ;Переходим к следующей итерации

ret

array:

dd 0

dd 0

dd 1

dd 2

**Задание 2:** написать программу, реализующую доступ к упакованному массиву с заданной разрядностью элементов: 3 бита

**Ход выполнения:**

1. Выбираем массив, элементы которого не превышают 7, например [0, 1, 2, 3, 64, 5, 6, 7]
2. Переводим числа в двоичную систему, а затем записываем биты от младшего к старшему.
3. Объединяем числа по 8 элементов и переводим в 16-ричную СС
4. Регистры для входных и выходных значений:

ESI – Адрес массива

AL, Bl – Элементы массива

1. Заносим программу в отладчик
2. Выполняем программу, проверяя каждый элемент поочерёдно, после каждого запуская убеждаясь, что программа работает корректно и результаты совпадают с полученными вручную.

**Код программы:**

LEA ESI,[a]

;извлекаем значение с индексом 0

MOV AL,[ESI]

MOV DL, 0x00

AND AL,DL

PRINT\_UDEC 1,AL

NEWLINE

;извлекаем значение с индексом 1

MOV AL,[ESI + 1]

MOV DL, 0x01

AND AL,DL

SHL AL,3

MOV BL,AL

MOV AL,[ESI]

MOV DL, 0x01

AND AL,DL

SHR AL,5

OR BL,AL

PRINT\_UDEC 1,BL

NEWLINE

;извлекаем значение с индексом 2

MOV AL,[ESI + 1]

MOV DL, 0x02

AND AL,DL

SHR AL,2

PRINT\_UDEC 1,AL

NEWLINE

;извлекаем значение с индексом 3

MOV AL,[ESI + 2]

MOV DL, 0x03

AND AL,DL

SHL AL,1

MOV BL,AL

MOV AL,[ESI + 1]

MOV DL, 0x03

AND AL,DL

SHR AL,7

OR BL,AL

PRINT\_UDEC 1,BL

NEWLINE

;извлекаем значение с индексом 4

MOV AL,[ESI + 3]

MOV DL, 0x04

AND AL,DL

SHL AL,4

MOV BL,AL

MOV AL,[ESI + 2]

MOV DL, 0x04

AND AL,DL

SHR AL,4

OR BL,AL

PRINT\_UDEC 1,BL

NEWLINE

;извлекаем значение с индексом 5

MOV AL,[ESI + 3]

MOV DL, 0x05

AND AL,DL

SHR AL,1

PRINT\_UDEC 1,AL

NEWLINE

;извлекаем значение с индексом 6

MOV AL,[ESI + 4]

MOV DL, 0x06

AND AL,DL

MOV BL,AL

MOV AL,[ESI + 3]

MOV DL, 0x06

AND AL,DL

SHR AL,4

OR BL,AL

PRINT\_UDEC 1,BL

NEWLINE

;извлекаем значение с индексом 7

MOV AL,[ESI + 4]

MOV DL, 0x07

AND AL,DL

SHR AL,3

PRINT\_UDEC 1,AL

NEWLINE

ret

section .data

;packed array [0, 1, 2, 3, 4, 5, 6, 7]

a: DB 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07

**Задание 3:**

Написать программу, реализующую алгоритм сортировки: сортировка вставками.

**Ход работы:**

1. Сортировка вставками заключается в том, что мы берем попавшееся нам число и вставляем на то место, на котором оно должно стоять в результате сортировки. В итоге все должно стоять на своем месте.
2. В качестве тестового массива возьмём [4,7,1,3,2,0,6,5], наша программа должна выдать [1,2,3,4,5,6,7,8]
3. Регистры для входных и выходных значений:

EBX – Адрес массива

ECX – Количество элементов массива

AL, DL – Элементы массива

EDI – Адрес массива

1. Заносим программу в отладчик
2. Выполняем программу и убеждаемся, что результат совпадает с вычисленным вручную.

**Код программы:**

xor eax, eax

LEA EBX, [a]

MOV ECX, 8

next:

PUSH EBX

PUSH ECX

MOV DL, [EBX]

MOV EDI, EBX

next1:

MOV AL, [EBX]

CMP AL, DL

JNL skip

MOV DL, AL

MOV EDI, EBX

skip:

INC EBX

LOOP next1

POP ECX

POP EBX

MOV AL, [EBX]

MOV [EDI], AL

MOV [EBX], DL

INC EBX

LOOP next

ret

section .data

a: DB 4, 7, 1, 3, 2, 0, 6, 5